2016 20th International Conference Information Visualisation

Evaluation of Sketchiness as a Visual Variable for 2.5D Treemaps

Daniel Limberger*, Carolin Fiedler, Sebastian Hahn, Matthias Trapp and Jürgen Döllner
Computer Graphics Systems Group
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany
*Email: *daniel.limberger@hpi.de, {firstname.lastname@hpi.de}*

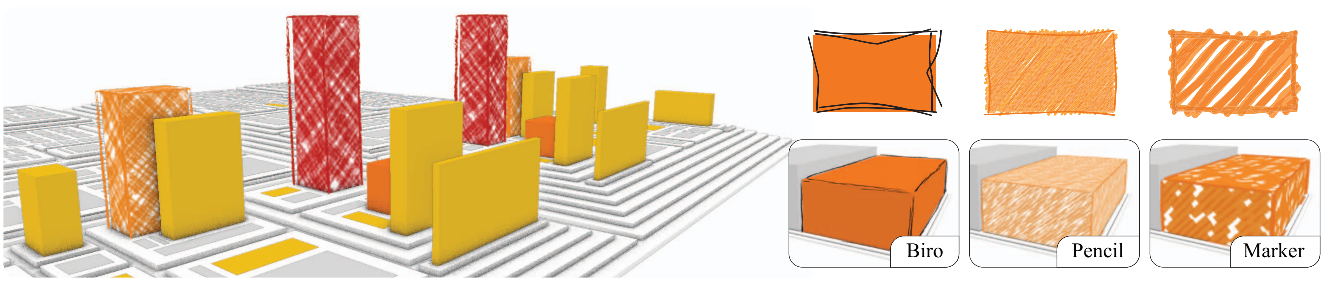


Figure 1. A 2.5D treemap using sketchiness to depict uncertainty for each leaf node of the treemap (left). Variations of sketchy rendering styles for 2.5D treemap blocks patterns (top) and example blocks (bottom).

***Abstract*—Treemaps serve as generic, effective tools to display, explore, and analyze multi-variate tree data in a scalable, interactive, and consistent way. In this paper, we discuss and evaluate sketchiness as visual variable of 2.5D treemaps. Sketchy rendering techniques allow us to map data, e.g., about uncertainty, imprecision, or vagueness, independently from mappings to other visual variables such as size, color, and height. To this end, we present a design space for sketchy rendering for 2.5D treemaps and corresponding implementation of a real-time sketchy rendering technique. The results of three user studies carried out indicate that sketchiness is a promising candidate for an independent visual variable for 2.5D treemaps, in particular to map ordinal data with a small range such as data that qualifies map items; it shows no strong interference with other visual variables such as color and height due to the regular gestalt of blocks and, hence, allows us to extend the expressiveness of 2.5D treemaps.**

***Keywords*-Treemap; Sketchiness; Visual Variable; Uncertainty; Visual Analytics; Real-time Rendering;**

I. Introduction

Treemaps provide mappings of tree-structured, hierarchical data by means of a space-constrained, recursively nested sets of rectangles that express the parent-child relationship among nodes and whose sizes are proportional to some computed per-node weights [1]. Data associated with nodes, here referred to as *attributes*, can be mapped by means of the *visual variables* of treemaps, e.g., size, color, texture, and shading [2].

2.5D treemaps extend treemaps by using the third dimension to extrude rectangles to *blocks* while still having the regular treemap's two-dimensional reference space and layout. Technically they are represented as interactive 3D models and often ambiguously denoted by *3D treemaps*. On the one hand, blocks facilitate emphasizing the hierarchical nesting of inner nodes and offer size and color as principal visual variables. On the other hand, blocks also provide another strong visual variable, namely height, which, in consequence, allows us to independently map three attributes (size, color, height) and to visually relate these attributes in a single 2.5D treemap view.

Many 2D and 2.5D treemap variants are applied in a growing number of applications and systems to visualize multivariate, large tree-structured data of, e.g., file systems [3], software systems [4], business data [5], stock markets [6], or controller performance data [7]. For example, visual software analytics tools build software maps based on 2.5D treemaps to provide insight into large software artifact hierarchies and their characteristics by relating code modules (nodes) with associated data about code metrics, software quality, development activity as well as budget and resource planning [8].

As Wood et al. [9] points out, *sketchiness* "offers different ways of communicating ideas of narrative, purpose, ownership, accuracy and aesthetic" and it "may be reliably used as a visual variable on an ordinal scale, but that caution should be exercised when representing interval or ratio scale data". Uncertainty, imprecision, or vagueness data, mostly represented by *ordinal variables with small range*, can be found in many treemap application scenarios. Of course, this kind of attribute data can be mapped to common visual variables. However, if we depict this data by means of sketchiness, which provides a fourth 'strong' visual variable for 2.5D treemaps, we thereby retain the other visual variables. Sketchiness also allows us to qualify

2375-0138/16 $31.00 © 2016 IEEE
DOI 10.1109/IV.2016.61

attribute mappings by mapping uncertainty in a generally understandable way. For example, in software maps we could superimpose the attribute ‘test coverage’ (sketchiness) to the mappings of lines-of-code (size), software complexity (height) and frequency of changes (color) to emphasize those modules that are difficult to comprehend, have been significantly altered recently *and* that are rarely tested. This paper provides the following contributions:

1) We discuss a design space of sketchiness for 2.5D treemaps with respect to rendering of sketchy outlines and surfaces, which is explained in detail for three distinct styles: Biro, (colored) Pencil, and Marker.
2) We evaluate sketchiness in terms of a visual 2.5D treemap variable and discuss its selective, associative, quantitative, as well as order and length characteristics.

II. Related Work

Related work includes findings in the fields of visualization of tree-structured and hierarchical data, non-photorealistic rendering (NPR), human perception, map design, and cartographic information display.

Since the original presentation of Johnsons and Shneiderman’s 2D treemaps [10] various treemap layouts have been investigated, improving layout readability, stability, and the graphical elements’ aspect ratio [11], [12]. 2.5D treemaps utilize the third dimension for additional information display based on the Cityscape metaphor [13] by transforming rectangular elements into stacked 3D blocks. Our GPU-based implementation follows the real-time rendering approach for 3D treemaps described in [14] that can efficiently process large amounts of tree data.

Sketchy rendering is often based on perturbing outlines (e.g., silhouettes) and modifying textures (e.g., hatches, strokes) in a sketchy way [15]. For example, if we map uncertainty values to edges, drawn in image space [16], we get sketchy outlines causing a kind of “shower door” effect. Alternatively, overlapping, edge-aligned and textured billboards can be placed in object space, enabling image-space independent sketchiness [17]. In this case, efficient and accurate line visibility can be achieved by GPU-based hidden line removal [18]. In this work, we follow the object-based approach for sketchy rendering. To draw textures in a sketchy style, we can hatch strokes over surfaces, e.g., via precomputed [19], [20] or recursive procedural Tonal Art Maps [21]. For hatching, a GPU-based, dynamic principal direction propagation emphasizes surface orientation [22] but does not significantly contribute to sketchiness of blocks in 2.5D treemaps due to their simple, regular shape. Most computer generated NPR styles come close to hand-drawn ones and seem to be preferred by users [23], [24].

Sketchiness in general allows us to influence how people read and understand depicted information [25]. Wood et al. [9] investigate sketchy rendering for information visualization and proposes a corresponding framework; we extend this work for the specific application of sketchiness in 2.5D treemaps. In particular, sketchiness can intentionally facilitate communication [26] or reduce the expected amount of cognitive processing due to its supposed simplicity [9]. It conveys visual imprecision, decreases people’s confidence in the underlying data quality and encourages constructive feedback, discussion, and participation [27], [28] as well as supports interactive exploration [29], [30].

In cartographic information display [31] and information visualization, visual variables provide a theoretical background for designing and understanding visual mappings [32]. Each visual variable (e.g., size, shape, color, texture) can be used to map one of the underlying multi-dimensional data but its choice depends on the variable’s characteristics including selective, associative, quantitative, length, and order properties [2], which significantly have impact on perception and understanding of the presented information.

Sketchiness can be viewed as visual variable that alters the way other visual variables are rendered, i.e., it perturbes their visual mappings. In particular, sketchiness appears to “have the capacity to carry information in its own right” [9] and allows us to map any ratio-scale data, while being most effective for ordinal data. For example, to depict uncertainty, Boukhelifa et al. [33] propose to use only 3 to 4 discrete line styles.

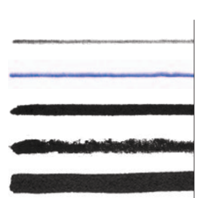
III. Design Space of Sketchy Rendering for 2.5D Treemaps

Sketchiness as a visual variable is accomplished in our approach by sketchy rendering applied to the components a 2.5D treemap, imitating hand-drawn outlines and fillings.

Different sketchy styles are implemented based on different NPR drawing concepts [34] such as pencils, colored pens, ballpoint pens, or markers (Fig. 3). As the degree of sketchiness can vary per block, we have to ensure that the sketchy rendering technique can be independently processed per block. In our approach, we focus on a small set of clearly distinguishable hand-drawn styles, in particular outlines and textures (Fig. 1) that fulfill this constraint.

A. Rendering of Sketchy Outlines

Sketchy outlines imitate lines that appear to be hand-drawn with a drawing tool. The essential characteristics are (1) graphical inconsistency (the stroke’s texture), (2) perturbation of spatial position (wobbling), and (3) overdraw (multiple strokes of varying intensity per line feature).



Graphical Inconsistency. Hand-drawn strokes can vary depending on line thickness, pressure—causing a sparser or denser texture grain and a reduced or increased color intensity respectively—as well as the type of drawing tool used. Common drawing tools for sketching

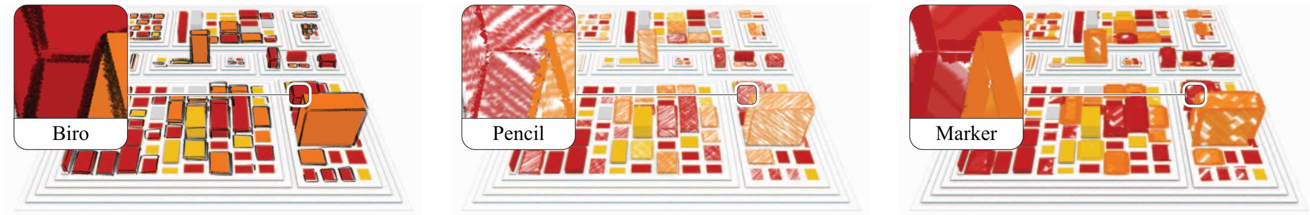
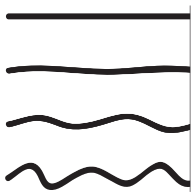


Figure 3. Different sketchy styles: Default shading with ballpen outlines (Biro), outlines imitating colored pencils combined with slightly desaturated hatching (Pencil), thick outlines and hatches for imitating text markers (Marker).

are (depicted top to bottom) graphite pencils, biros, sharpies, crayons, and markers. The variations in grain, intensity, and thickness can be accounted for by using either image-based or procedurally-generated textures for real-time rendering.



Spatial Position. The appearance of strokes is also determined by their straightness, which can be influenced due shaking or wobbling (*perturbation*) by noise. We use the maximum perturbation of a stroke as a control for the intensity of sketchiness [33]. For real-time rendering, 3D spatial perturbation is implemented by subdividing strokes (*tessellation*) into multiple line segments and individually displacing each point by a spatially referenced offset implemented by a 3D noise texture look-up. With the amplitude, frequency (or number of octaves), and type of noise, the appearance of single strokes can be further refined. For example, white noise (high frequency) creates a highly unsettled, edgy look, while simplex noise causes a rather smooth and curvy appearance. In contrast to most image-based approaches, our method is view-independent and produces no shower door effects.

Overdraw. The display of a single line is not limited to a single stroke. Instead, multiple strokes of the same texture but with varying perturbation can be applied (*overdraw*). We achieve this by shifting the 3D noise look-up randomly for every stroke. Although an overdraw of strokes with varying intensities (e.g., due to uneven pressure applied when drawing) allows us to increase the visualizations aesthetics, we do not consider it to significantly affect the perceived degree of sketchiness of outlines here.

Implementation.: Sketchy outlines in 2.5D treemaps are rendered as follows: For all edges of all sketchy blocks an additional draw call (line draw) is dispatched. The edges are GPU-tessellated and displaced per vertex using spatially referenced 3D offsets. For the offsets, a single tileable, linear filtered 3D noise texture is pre-computed and scaled appropriately. Finally, view-oriented billboards are derived along the line segments and textured via distance transformed stroke maps. Since the number of leafs comprised by treemaps strongly varies, a level-of-detail concept as well as an appropriate effect (re)parameterization in terms of scale has not been investigated so far.

B. Rendering of Sketchy Textures

To render block surfaces in a sketchy style, we use hatch textures to modify the surface appearance such as the Pencil and Marker examples in Fig. 3. The design space for sketchy textures is similar to sketchy outlines; it includes filling style, stroke texture and stroke density.

As a replacement of solid shading in the block rendering technique, either hatching or stippling can be used. Both convey texture, tone, and form by drawing closely spaced dapples or parallel as well as orthogonal strokes or stipples. For every stroke of a texture, a homogeneous parameterization of graphical inconsistency and spatial perturbation is implied in order to maintain the blocks distinct appearance. Varying densities of strokes might cause quite different effects—a cross-hatched surface displays the original color much better than a sparse hatched surface since the sparse version contains a lot more background color. To retain the thematic mapping, hue hatching is typically used for sketchiness in information visualization [9]; instead of depicting lighting and shading information, the hatching intensity indicates the degree of sketchiness (Fig. 4).

Implementation.: For real-time rendering using hatching we use pre-rendered distance transformed Tonal Art Maps (*TAMs*) [19]. A TAM is a set of texture images of different stroke densities, i.e., hatching intensities, that get blended appropriately during fragment shading. Here, shower door effects are avoided by stroke-aware mipmapping. In the case of strongly discretized color mapping—we prefer five buckets [33]—a five-level sketchiness allows

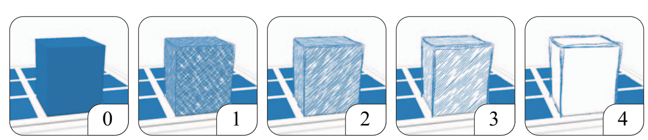


Figure 4. Sketchiness based on colored pencil drawing: a block can be rendered in a range from non-sketchy (left) to maximal sketchy (right). For it, the hatching intensity is designed to decrease evenly.

us to slightly account for shading without compromising its discernibility. Similar to the setup for sketchy outlines, the TAMs' size and stroke-scales should be adjusted to the treemap size.

IV. Evaluation

For the evaluation of the three sketchy rendering styles *Biro*, colored *Pencil*, and *Marker* (Fig. 3) we conducted a series of experiments to test their usability as a visual variable in 2.5D treemaps with focus on three aspects:

Pre-study Does sketchiness as a visual variable affect the perceived height of 2.5D graphical elements in a similar way as the perceived area of 2D elements [9]?

Study I Can the three distinct styles be processed pre-attentively (*selective* characteristic)?

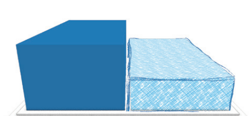
Study II Are users capable of recognizing an *order* when different degrees of sketchiness are depicted?

A. Effect on Perceived Height (Pre-Study)

Since there is only sparse information on the height perception of sketched graphical elements in a 2.5D reference space, we tested for a significant effect on the height estimation of individual treemap elements when sketchiness is used (as shown for relative size estimation in 2D [9]).

Participants. We conducted a study with 50 participants (13 female, 37 male) aged 20 to 53 ($M = 27.4$, $SD = 7.2$) with an increased daily desktop computer usage ($M = 8.1$ hours, $SD = 2.7$) and most either strongly agreed ($n = 15$) or strongly disagreed ($n = 17$) in beeing experts in computer graphics and visualization on a five-value Likert scale.

Procedure. The experiment was implemented and executed as an online survey; the rendering styles were randomly assigned, each style having a balanced number of answers. First, participants where asked for their demographic information including age, gender, and PC usage per day. Next, two pre-answered examples showing a reference block with a higher and a shorter block in the same non-sketchy style were introduced to every participant.



Then, images of two blocks, one reference non-sketchy block (left) with a constant height of 100 height units (*hu*) and the other (right) faced with the study conditions, were shown. The users were asked to estimate the height of the block on the right. The input was restricted to a height range of 1 to 200. During the study the conditions of the actual block's height, the point of views' angle with a constant distance to the treemap center, and the rendering style varied.

Design. The experiment was a 3 × 4 × 4 mixed design. Two independent variables, angle and actual height, were tested within-subjects, while the rendering styles were tested between-subjects as follows: *Angle* (0°, 15°, 30°), *ItemHeight* (36, 77, 132, 187), and *Style* (Solid, Biro, Pencil, Marker). The conditions were fully randomized for each user to prevent order and learning effects. Aside from training, the amount of estimations were 50 participants × 3 angles × 4 heights = 600. Each estimation results in an absolute error of the estimated height $AbsErr = |ItemHeight - EstimatedHeight|$, log-transformed for better readability via $log_2(1 + AbsErr)$.

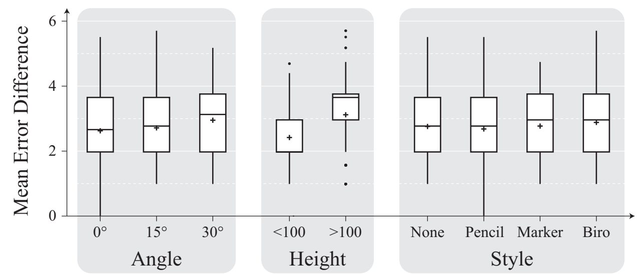


Figure 5. Pre-study results depicting the mean error differences for (f.l.t.r.) angle, height, and style groups.

Results. We declared items as in-acceptable outliers, if the estimated height was above 100hu as the actual block's height was below 100hu and vice versa. With this constraint 5 estimates (0.01%) were removed from the overall trials. As expected, changing the camera angle results in significant differences (Fig. 5, Angle) in the estimation error ($F_{2,76} = 17.19$, $p < .0001$). A post-hoc comparison showed that the angles 0° and 15° yield lower errors by 12% ($p < .0001$) and 8% ($p = .003$) respectively compared to an angle of 30°. The error rates with respect to the actual block heights showed significant differences: $F_{3,115} = 54.52$, $p < .0001$. Besides the comparison of 36hu and 132hu ($p = .999$) all differences between pairs were statistically significant in a post-hoc test (all $p < .001$). In addition to looking at the four different sampled heights, we split the heights into two groups, one lower and one higher than 100hu. A one-factorial Anova showed significant differences between these groups: $F_{1,49} = 72.59$, $p < .0001$ (Fig. 5, Height). Finally, for the varying rendering techniques no significant difference for the height estimation errors was found: $F_{3,43} = .80$, $p < .50$ (Fig. 5, Style).

B. Pre-Attentive Processing (Study I)

Since Biro had a bad performance in early tests—especially under the assumption of equality, i.e., all elements having outlines—we focused on Marker and Pencil for pre-attentive perception and in addition, differentiated between low and high sketchiness (Fig. 4, 1 and 3).

Participants. This experiment was conducted with 24 participants (4 female, 20 male) aged 20 to 33 ($M = 24.9$, $SD = 3.4$) and having diverse background knowledge in the field of 2.5D treemaps and visualization in general.

Procedure. Every participant had to perform 3 tasks (color, sketchiness, conjunction) for a specific rendering style (Marker or Pencil). We used an eye-tracking setup to ensure that participants had to focus on a fixation cross for a time of 700ms before an image with the used visual variable was shown for maximum 200ms with a stimulus present or absent. After each image, users were asked whether they recognized an element that was distinct to all others in color, sketchiness, or a conjunction of both.

Apparatus. We used an eye-tracking device (EyeFollower[1] by InteractiveMinds) that allows for accurate gaze-tracking within natural head movements at a desktop environment. The participant was placed about 60cm away from a 24inch monitor with a FullHD screen resolution (1080p). Additionally, an observation monitor was placed behind a partition wall that allowed the observer to check for losses of head-tracking during the experiment. The used software recorded gaze sample data with 120Hz.

Design. The experiment was a $3 \times 2 \times 2 \times 2$ mixed design with three independent variables (size, presence, sketchiness value) tested within-subjects, while the rendering styles were tested between-subjects: *Size* (4×4, 7×7, 10×10), *Presence* (absent, present), *Sketchiness* (low, high), and rendering *Style* (Pencil, Marker). The trials started with the color task, followed by the sketchiness part and the conjunction of both. Again, we fully randomized the variables tested within subject to prevent order and learning effects for *Size*, *Presence*, and *Sketchiness* for each user. All in all, 1728 images were shown for 24 (participants) $\times$ 3 (sizes) $\times$ 2 (present/absent) $\times$ 2 (sketchiness values) $\times$ 3 (positions) $\times$ 3 (tasks, color/sketchiness/conjunction).

Results. The results discussed are enlisted in Table I and shown in detail in Fig. 6. The pre-attentive processing of the visual variable color that serves as a comparative measure for our results showed a very high accuracy as well as precision and recall. While the trials from the task of pre-attentively processing the implemented sketchiness styles also scored high values, as expected, the conjunction of both performed poorly. Fisher's exact test significantly showed the independence by tasks ($p < .0005$ for I.TB, I.TS, and I.TC). A comparison of high and low *Sketchiness* showed significant differences ($p = .007$ for I.SL and I.SH) with a better accuracy for high sketchiness. Looking at the different styles, we found a significant difference in the sketchiness task ($p = .0002$ for I.SP and I.SM) with Pencil outperforming Marker in accuracy. Furthermore, a significant difference between the used sizes ($p < .0001$ for I.S4 and I.SX) was found: the participants ability of pre-attentively processing sketchiness decreases as the number of elements increases.

[1]http://www.interactive-minds.com/eye-tracker/eyefollower

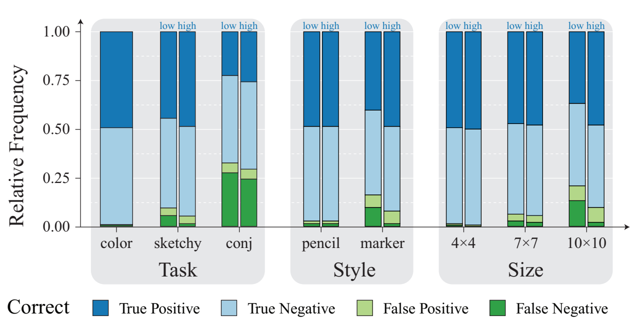


Figure 6. Results of Study I depicting the relative frequency for (f.l.t.r.) task, style, and size (low and high sketchiness).

Table I
RESULTS FOR PRE-ATTENTIVE PROCESSING (STUDY I).

#	Variable	Value	AC	PC	RC
I.TB	*Task*	color/baseline	.991	.995	.986
I.TS	*Task*	sketchiness	.927	.959	.931
I.TC	*Task*	conjunction	.620	.904	.481
I.SL	*Sketchiness*	low	.905	.919	.889
I.SH	*Sketchiness*	high	.947	.925	.972
I.SP	*Style*	pencil	.927	.986	.972
I.SM	*Style*	marker	.883	.932	.889
I.S4	*Size*	4×4	.991	.993	.993
I.S7	*Size*	7×7	.965	.951	.944
I.SX	*Size*	10×10	.847	.917	.847

C. Order of Elements (Study II)

Boukhelifa et al. stated that the number of distinct levels for a sketchy rendering style is about 3 to 4 [33]. We implemented these distinct levels for all styles and conducted another study on the ability of participants to bring these levels into correct order (based on the sketchiness value). The study was again performed as an online survey with 116 participants from which 91 gave complete answers.

Procedure and Design. Each survey included 6 sets of 5 images: an image depicting a distinct sketchiness value from non-sketchy to full sketchy (see Fig. 4) for each of the three sketchy rendering styles of two different sizes. The experiment was a 2×3 within-subject design: *Size* (*coarse*: 3×3, *fine*: 9×9) and rendering *Style* (pencil, marker). The number of images shown were for 91 participants $\times$ 2 sizes $\times$ 3 sketchy rendering styles = 546 tests.

Results. For set of images a score was calculated (based on the order) representing the number of correct answered positions, with a range from 0 (all positions wrong) to 5 (completely correct). A Kruskal-Wallis test showed significant differences between the rendering styles performance ($p < .001$, Fig. 7). A post-hoc comparison showed signifi-

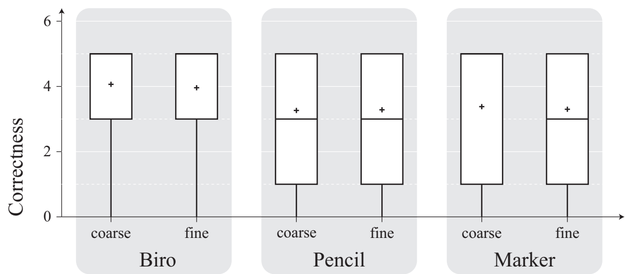


Figure 7. Study II results depicting correctness for (f.l.t.r.) Biro, Pencil, and Marker, each in coarse and fine variation.

cant differences between Biro and Marker ($p = .00032$) as well as Biro and Pencil ($p = .00015$). However, we could not find this effect in the comparison between Pencil and Marker ($p = .72$). The Biro style also achieved the highest score ($M = 4.00$, $SD = 1.66$), while Marker ($M = 3.32$, $SD = 1.82$) and Pencil ($M = 3.26$, $SD = 1.79$) achieved similar lower scores. Also, we could not find a significant difference between different sizes of images ($p < .001$).

V. Conclusions and Future Work

Our experiments show that sketchiness can be used as independent visual variable to map ordinal data in 2.5D treemaps without having impact on reading and interpreting other visual variables such as color and height. The general weakness of sketchy rendering, the lowever data/ink ratio, is less relevant due to the uniform and regular block shape used by 2.5 treemaps—there is no noticable "chart junk" [35] introduced. Additionally, we could measure a high accuracy for the selective pre-attentive processing for 2 of 3 styles (Marker, Pencil). The use of a conjunction of sketchiness and color is poorly processed pre-attentively as expected. In addition, the pure use of outlines (as in the biro style) leads to poor pre-attentive processing and as the size of elements increases, the ability to pre-attentively process the items decreases. With the length of the design space of 3 to 4 distinct values as described by Boukhelifa et al., we could also show that users are able to bring depicted elements in an order based on their sketchiness value for all styles. In summary, the expressiveness of 2.5D treemaps can be extended by sketchiness as visual variable on its own right if applied to appropriate data such as ordinal data with small range.

In future work, we would like to explore how time-varying uncertainty data can be mapped and animated in 2.5D treemaps, e.g., to explore changes over time. We also want to explore how sketchy outlines can be integrated into a global illumination concept for 2.5D treemaps (e.g., shadow casting) and how sketchy line patterns for general ordinal data not related to uncertainty could be look like.

Acknowledgment

This work was funded by the Federal Ministry of Education and Research (BMBF), Germany, within the InnoProfile Transfer research group "4DnD-Vis" (www.4dndvis.de).

References

[1] B. Shneiderman and C. Plaisant, "Treemaps for space-constrained visualization of hierarchies," Web document, 2009. [Online]. Available: http://www.cs.umd.edu/hcil/treemap-history/index.shtml

[2] M. Carpendale, "Considering visual variables as a basis for information visualisation," 2003.

[3] B. Shneiderman, "Tree visualization with treemaps: A 2D space-filling approach," *ACM Trans. Graph.*, vol. 11, no. 1, pp. 92–99, 1992.

[4] R. Wettel and M. Lanza, "CodeCity: 3D visualization of large-scale software," in *Proc. ACM ICSE Companion*, 2008, pp. 921–922.

[5] R. Vliegen, J. J. van Wijk, and E.-J. van der Linden, "Visualizing business data with generalized treemaps," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 789–796, 2006.

[6] M. Wattenberg, "Visualizing the stock market," in *Proc. ACM CHI EA*, 1999, pp. 188–189.

[7] W. Mitchell, D. Shook, and S. L. Shah, "A picture worth a thousand control loops: An innovative way of visualizing controller performance data," in *Invited Plenary Presentation, Control Systems*, 2004.

[8] J. Bohnet and J. Döllner, "Monitoring code quality and development activity by software Maps," in *Proc. ACM MTD*, 2011, pp. 9–16.

[9] J. Wood, P. Isenberg, T. Isenberg, J. Dykes, N. Boukhelifa, and A. Slingsby, "Sketchy rendering for information visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2749–2758, 2012.

[10] B. Johnson and B. Shneiderman, "Treemaps: A space-filling approach to the visualization of hierarchical information structures," in *Proc. IEEE VIS*, 1991, pp. 284–291.

[11] M. Bruls, K. Huizing, and J. van Wijk, "Squarified treemaps," in *IEEE TCVG Symposium on Visualization*, 1999, pp. 33–42.

[12] B. B. Bederson, B. Shneiderman, and M. Wattenberg, "Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies," *ACM Trans. Graph.*, vol. 21, no. 4, pp. 833–854, 2002.

[13] C. Keskin and V. Vogelmann, "Effective visualization of hierarchical graphs with the cityscape metaphor," in *In Workshop on NPIV*, 1997, pp. 13–14.

[14] M. Trapp, S. Schmechel, and J. Döllner, "Interactive rendering of complex 3D-treemaps with a comparative performance evaluations," in *GRAPP & IVAPP 2013*, 2013, pp. 165–175.

[15] T. Strothotte and S. Schlechtweg, *Non-photorealistic computer graphics: modeling, rendering, and animation*. Morgan Kaufmann, 2002.

[16] M. Nienhaus and J. Döllner, "Sketchy drawings," in *Proc. ACM AFRIGRAPH*, 2004, pp. 73–81.

[17] J. Döllner and M. Walther, "Real-time expressive rendering of city models," in *IV*, 2003.

[18] F. Cole and A. Finkelstein, "Fast high-quality line visibility," in *Proc. I3D*. ACM, 2009, pp. 115–120.

[19] E. Praun, H. Hoppe, M. Webb, and A. Finkelstein, "Real-time hatching," in *Proc. ACM SIGGRAPH*, 2001, pp. 581–.

[20] M. Webb, E. Praun, A. Finkelstein, and H. Hoppe, "Fine tone control in hardware hatching," in *Proc. ACM NPAR*, 2002, pp. 53–ff.

[21] L. Szécsi and M. Szirányi, "Recursive procedural tonal art maps," pp. 57–66, 2014.

[22] Y. Kim, J. Yu, X. Yu, and S. Lee, "Line-art illustration of dynamic and specular surfaces," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 156:1–156:10, 2008.

[23] F. Cole, A. Golovinskiy, A. Limpaecher, H. S. Barros, A. Finkelstein, T. Funkhouser, and S. Rusinkiewicz, "Where do people draw lines?" in *ACM SIGGRAPH 2008 Papers*, 2008, pp. 88:1–88:11.

[24] T. Isenberg, P. Neumann, S. Carpendale, M. C. Sousa, and J. A. Jorge, "Non-photorealistic rendering in context: An observational study," in *Proc. ACM NPAR*, 2006, pp. 115–126.

[25] T. Isenberg, "Evaluating and validating non-photorealistic and illustrative rendering," in *Image and video-based artistic stylisation*. Springer, 2013, pp. 311–331.

[26] J. Walny, S. Carpendale, N. Henry Riche, G. Venolia, and P. Fawcett, "Visual thinking in action: Visualizations as used on whiteboards," *IEEE Trans. Vis. Comp. Graph.*, vol. 17, no. 12, pp. 2508–2517, 2011.

[27] S. Zanola, S. Fabrikant, and A. Çöltekin, *The Effect of Realism on the Confidence in Spatial Data Quality in Stereoscopic 3D Displays: (refereed Extended Abstract)*. Geographisches Institut, 2009.

[28] M. Luboschik, A. Radloff, and H. Schumann, "Poster: Using non-photorealistic rendering techniques for the visualization of uncertainty," 2010.

[29] S. Greenberg, S. Carpendale, N. Marquardt, and B. Buxton, *Sketching user experiences: The workbook*. Elsevier, 2011.

[30] J. Browne, B. Lee, S. Carpendale, N. Riche, and T. Sherwood, "Data analysis on interactive whiteboards through sketch-based interaction," in *Proc. ACM ITS*, 2011, pp. 154–157.

[31] J. Bertin, *Semiology of Graphics*. University of Wisconsin Press, 1983.

[32] A. Skupin, "From metaphor to method: Cartographic perspectives on information visualization," in *Proc. IEEE InfoVis*, 2000, pp. 91–97.

[33] N. Boukhelifa, A. Bezerianos, T. Isenberg, and J.-D. Fekete, "Evaluating sketchiness as a visual variable for the depiction of qualitative uncertainty," *IEEE Trans. Vis. Comp. Graph.*, vol. 18, no. 12, pp. 2769–2778, 2012.

[34] S. Schlechtweg and A. Raab, "Rendering line drawings for illustrative purposes," in *Computational Visualization: Graphics, Abstraction, and Interactivity*, T. Strothotte, Ed. Springer Verlag, 1998, pp. 65–89.

[35] E. R. Tufte, *The Visual Display of Quantitative Information*. Cheshire, CT, USA: Graphics Press, 1986.